Federated Learning Over Wireless Channels: Dynamic Resource Allocation and Task Scheduling

Shunfeng Chu, Jun Li, *Senior Member, IEEE*, Jianxin Wang, Zhe Wang, *Member, IEEE*, Ming Ding, *Senior Member, IEEE*, Yijin Zhang, *Senior Member, IEEE*, Yuwen Qian, and Wen Chen, *Senior Member, IEEE*

***Abstract*—With the development of federated learning (FL), mobile devices (MDs) are able to train their local models with private data and send them to a central server for aggregation, thereby preventing leakage of sensitive raw data. In this paper, we aim to improve the training performance of FL systems in the context of wireless channels and stochastic energy arrivals of each MD. To this purpose, we dynamically optimize MDs' transmission power and training task scheduling. We first model this dynamic programming problem as a constrained Markov decision process (CMDP). Due to high dimensions of the proposed CMDP problem, we propose online stochastic learning methods to simplify the CMDP and design online algorithms to obtain an efficient policy for all MDs. Since there are long-term constraints in our CMDP, we utilize a Lagrange multipliers approach to tackle this issue. Furthermore, we prove the convergence of the proposed online stochastic learning algorithm. Numerical results indicate that the proposed algorithms can achieve better performance than the benchmark algorithms.**

***Index Terms*—Federated learning, Markov decision processes, stochastic learning, resource allocation, dynamic programming.**

I. Introduction

IN THE last decade, we have witnessed a series of amazing breakthroughs, such as AlphaGo, machine learning and artificial intelligence (AI), which have become the most cutting-edge technology in both academia and industry communities [1]. Distributed machine learning based on mobile edge computing (MEC) of the wireless networks is also one of the current hot research directions [2]. The sample data for machine learning can be obtained by collecting massive amounts of data from mobile devices (MD) distributed in the wireless network. By training the local data, the training performance of machine learning can be greatly improved.

Although offloading the local sample data of distributed MDs for centralized learning significantly improves the performance of machine learning, this mechanism suffers from two flaws. First, transmission delays from distributed MDs to the central cloud via backbone network are extremely large. Second, the local data often contains the private information of MDs, and uploading the private information to the central cloud will lead to the risk of personal privacy leakage. To cope with these two issues, federated learning (FL) has been introduced to act as an emerging distributed machine learning paradigm for MEC networks. In this manner, MDs train their local data and send their local model updates to a task publisher iteratively instead of uploading the raw data to a central server [3], [4], which brings the following two benefits in general. First, the communication latency and the energy consumption for computation can be significantly reduced owing to the fact that MDs are not required to upload huge amounts of local data for training to an edge server. Second, MDs upload their local model instead of the raw data to the edge server, which greatly reduces the risk of personal privacy information leakage [5].

Despite the aforementioned advantages of FL, there are still many challenges that have not been solved until now. Some existing studies [6], [7] adopted an idealized assumption that all MDs participating in FL are immune to the wireless and computation resource constraints. Reference [8] only focused on a practical Federated-Averaging algorithm for distributed DNN training and the training performance. Many studies [9], [10], [11], [12] have been committed to further reducing the communication overhead by developing compression methods. However in practice, MDs usually suffer from energy consumption constraints that may reduce the network lifetime and training efficiency.

In addition, frequent wireless communication is usually required for uploading and downloading the model parameters, which would increase the communication overhead and the training latency [13]. Therefore, it is necessary to design a feasible resource scheduling scheme to improve the communication and energy efficiency of FL. A number of existing works have studied the important problems related to the implementation of FL over wireless networks.

Manuscript received 28 September 2021; revised 18 June 2022; accepted 26 July 2022. Date of publication 3 August 2022; date of current version 9 December 2022. This work was supported in part by National key project 2020YFB1807700 and 2018YFB1801102, in part by the National Natural Science Foundation of China under Grants 61872184, 62071236 and 62071296, in part by the Fundamental Research Funds for the Central Universities of China with No. 30921013104 and No. 30920021127, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20210331, in part by Future Network Grant of Provincial Education Board in Jiangsu, and in part by Shanghai Kewei 20JC1416502 and 22JC1404000. The associate editor coordinating the review of this article and approving it for publication was K. Zeng. *(Corresponding authors: Jun Li; Zhe Wang.)*

Shunfeng Chu, Jun Li, Jianxin Wang, Yijin Zhang, and Yuwen Qian are with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: shunfeng.chu@njust.edu.cn; jun.li@njust.edu.cn; wangjxin@njust.edu.cn; yijin.zhang@gmail.com; admon@njust.edu.cn).

Zhe Wang is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: zwang@njust.edu.cn).

Ming Ding is with the Data61, CSIRO, Sydney, NSW 2601, Australia (e-mail: ming.ding@data61.csiro.au).

Wen Chen is with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wenchen@sjtu.edu.cn).

Digital Object Identifier 10.1109/TCCN.2022.3196009

In [14], two update methods are developed to reduce the uplink communication costs for FL. In [15], the joint power and resource allocation is studied for achieving ultra-reliable low latency communication in vehicular networks. The work in [16] proposed a new approach to minimize the computing and transmission delay for FL algorithms. Reference [17] developed a novel framework that enables the implementation of FL algorithms over wireless networks and formulated an optimization problem that jointly considers user selection and resource allocation for the minimization of FL training loss. Without taking into account of the energy constraints and battery dynamics of MDs, [18] formulated a FL over a wireless networks as a static optimization problem, and exploited the problem structure to decompose it into three static convex sub-problems. The work [19] proposed a static scheduling scheme to efficiently execute distributed learning tasks in an asynchronous manner while minimizing the gradient staleness on wireless edge nodes with heterogeneous computing and communication capacities. For energy efficient FL over wireless networks, [20] studied a joint learning and communication problem with the goal of minimizing the total energy consumption of the system under a latency constraint in FL system. The work in [21] introduced an energy-efficient strategy for bandwidth allocation under learning performance constraints. In [22], two optimization problems concerning the learning performance and the energy consumption of the workers were formulated and solved for appropriate local processing and communication parameter configuration. Reference [23] proposed a novel joint dataset and computation management scheme that jointly optimizes the amount of dataset and computation resources to balance the learning efficiency and energy consumption in FL system. However, the MEC networks are usually time-varying in the dynamic process and thus the methods in [18], [19] will result in considerable performance loss. The work [24], [25] model the channel and energy dynamically, and exploit the dynamic scheduling algorithm to obtain the asymptotically optimal results. Thus, it is of vital importance to develop efficient dynamic resource scheduling schemes to improve the performance of FL.

In this paper, we utilize constrained Markov decision processes (CMDP) as a mathematical tool to obtain an optimal algorithm for dynamic resource scheduling for FL, where each MD send local model updates trained on their local raw data iteratively to a common edge server, and the edge server aggregates the parameters from MDs participating in local training and broadcasts the aggregated parameters to all the MDs. In particular, each MD possesses computing units with computing capability, which can be used for local machine learning with local raw data. In order to improve the training performance of FL,[1] we propose an efficient stochastic optimization algorithm for scheduling resources of MDs in the FL processes by optimizing the size of raw data for local training and the transmit power of MDs to upload the local model.

Our main contributions are listed as follows.

1) Due to the dynamic nature of wireless network and battery status of MDs, we consider resource scheduling of FL in dynamic scenarios. Thus, we model the resource scheduling problem of the FL process as a CMDP problem, and improve the performance of FL by optimizing the size of the local training data at the MD side.
2) Since the state-action space dimension in the CMDP problem is relatively large and there are a few constraints in the dynamic problem, we simplify the stochastic optimization problem by proving an equivalent Bellman equation and using the Lagrange multipliers method.
3) We use approximate MDP and stochastic learning methods to analyze the CMDP problem, and design centralized online algorithms to obtain resource scheduling policy for all MDs. Besides, we provide effective analysis for the convergence of the online stochastic learning algorithms.

Although the idea of applying CMDP to design dynamic resource scheduling is not new, we are motivated to address the resource scheduling issues of resource constraints and dynamics of FL. To achieve high-quality learning performance, a reasonable constrained dynamic scene is essential to the resource scheduling issues [27], [28], [29]. Previous work has utilized CMDP as a mathematical model to design effective algorithms for resource scheduling in wireless networks [13], [30], [31], which is considered as an effective tool for solving dynamic and temporal-correlated problems. Inspired by this, we apply CMDP as the mathematical scene to address the resource scheduling problem in the FL process. Nevertheless, previous work still has some shortcomings in solving CMDP problems. The literature [13] adopted a deep learning algorithm that allows the edge server to learn and find optimal decisions without any a priori knowledge of network dynamics in the CMDP. However, reinforcement learning (e.g., Deep Q-learning [13]) is poorly scalable and requires a lot of computing power and time for training. The literature [30] solved the CMDP offloading problem by linear programming and Q-learning method, which have high spatial complexity. The work [31] developed a threshold-based algorithm to obtain the optimal delay-power tradeoff efficiently, in which the authors used the special structure of the mathematical model to solve the CMDP problem.

The rest of this paper is organized as follows. Section II describes the system model and dynamic analysis. CMDP-based dynamic resource scheduling problem is formulated in Section III. Section IV proposes approximate MDP and stochastic learning methods to simplify the CMDP problem, and designs online algorithms to obtain an efficient policy. Section V presents the simulation results. Finally, Section VI concludes this paper.

II. System Model

Consider a wireless synchronous FL system consisting of an edge server and N MDs as shown in Fig. 1. Each MD

[1] In general FL, existing work [26] used the accuracy of the test set to measure the performance of machine learning after training. Since the quantitative analysis of the accuracy in the test set is relatively difficult, we use the size of local dataset accumulated from MDs over iterations to evaluate the accuracy of the machine learning model [13].

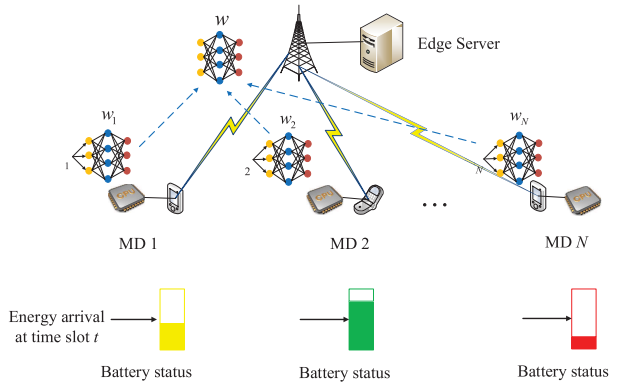


Fig. 1. The wireless federated learning network.

is equipped with computing and energy harvesting modules. Having access to a vast range of local data, each MD is able to train the machine learning model locally using the harvested energy from the environment. In most circumstances, each MD's local datasets have two cases: independently and identically distributed (IID) and non-IID. In [29], the authors investigated how to improve the performance of FL on non-IID datasets by model-free reinforcement learning method. However, it is difficult to quantify the performance of non-IID FL in dynamic processes and the computational complexity of model-free algorithms is typically high. In order to reduce the computational complexity of scheduling algorithms in dynamic environments, we adopt IID data samples for local training in our system model. To improve the model training efficiency and protect data privacy, the FL technique is adopted as an iterative model updating process between the edge server and MDs.

We first briefly introduce the main procedures as follows. In each learning iteration t, the n-th MD selected by the edge server for parameter update first selects $b_n(t)$ bits of training data from the local data set, where the size of selected data is determined by the edge server according to the energy status of the MD, i.e., the battery energy at the beginning of this iteration. At the beginning of each iteration, each mobile device sends its energy status information to the edge server. Because of the extremely small amount of data (usually several bits), the transmission delay and energy consumption of the energy state are often negligible. Hence, we assume that the edge server knows the energy status of all MDs in advance at the beginning of the learning iteration. Then, each MD performs local training and obtains the local model parameters. Due to the strict synchronous FL setting, MDs receive the most recent global model from the edge server at the start of each time slot and must upload the local models to the edge server on time. Afterwards, the n-th MD who has been selected by the edge server transmits the parameters to the edge server in the uplink using the power of $P_n(t)$ according to MDs' remaining energy and channel state. Finally, the edge server aggregates the local training parameters from all the participating MDs and then it broadcasts the updated global training parameters (e.g., weighted average over the local parameters) back to all the MDs. At the end of this iteration, each MD opportunistically harvests energy from the environment and stores the energy in the rechargeable battery. The above process is repeated until the learning model reaches the desired accuracy level.

In the following subsections, we will explain the above process in more details. We mainly discuss each learning iteration in three stages: dynamic energy harvesting, local model training, model parameter transmission and aggregation.

A. Dynamic Energy Harvesting

We assume that the n-th MD is equipped with a rechargeable battery with a limited capacity of $E_n^{\max}$.[2] At the beginning of each iteration t, we denote the n-th MD's energy state as $E_n^{\mathrm{sta}}(t)$ which is the remaining energy carried from the previous iteration. According to its energy state, the edge server decides whether or not to proceed to local model training and uplink parameters transmission for each MD. We let $E_n^{\mathrm{cop}}(t)$ denote the n-th MD's computation energy for local model training and $E_n^{\mathrm{com}}(t)$ denote the communication energy for parameters transmission, respectively, which will be discussed in more details in the next two subsections. Note that each MD's energy consumption cannot exceed the energy state in this iteration. At the end of the iteration t, we consider that each MD is able to harvest energy from the environment and store the energy in the rechargeable battery. We denote the energy harvesting process for the n-th MD by $\{\boldsymbol{E}_n^{\mathrm{arr}}(\cdot) : n \in \mathcal{N}\}$, which follows an independent stationary Poisson distribution with average arrival rate $\mathbb{E}[E_n^{\mathrm{arr}}] = \lambda_n$ [33].

Similar to [34], the energy state of the n-th MD at the beginning of iteration $t+1$ can be updated by the following recursion, i.e.,

$$E_n^{\mathrm{sta}}(t+1) = \min\Big\{\big[E_n^{\mathrm{sta}}(t) - \lceil E_n^{\mathrm{com}}(t) + E_n^{\mathrm{cop}}(t)\rceil\big]^+ + E_n^{\mathrm{arr}}(t), E_n^{\max}\Big\}, \quad t \geq 1, \tag{1}$$

where $\lceil\cdot\rceil$ denotes the ceiling operator, which is similar to an integer multiple of energy packs in [34], and $x^+ \triangleq \max\{x, 0\}$.

B. Local Model Training

At the beginning of each learning iteration t, each MD first selects $b_n(t)$ bits of data samples from the local dataset to perform a machine learning algorithm, and then it obtains the local model parameters. Intuitively, the choice of b_n depends on the available energy in its battery.[3] The MD can train a larger size of the training data if it has more sufficient battery energy in the current iteration. Otherwise, it trains less data or takes no training for this iteration. We assume it consumes C_n CPU cycles to train a unit sampled data on the n-th MD. The CPU frequency, denoted by f_n (in CPU cycle/s), is considered as a measurement of computation capacity of the n-th MD.

[2]For the ease of analysis, we quantize the battery capacity in to $E_n^{\max}+1$ uniform levels $\{0, 1, \ldots, E_n^{\max}\}$ [32].

[3]Since the size of a single sample is relatively small, we assume that $b_n(t)$ is a continuous variable.

In iteration t, the processing time of local training on the n-th MD is given by

$$\tau_n^{\mathrm{cop}}(t) = \frac{b_n(t)C_n}{f_n}. \tag{2}$$

According to [18], the computation energy $E_n^{\mathrm{cop}}(t)$ consumed by local training of the n-th MD in the iteration t is given by

$$E_n^{\mathrm{cop}}(t) = \alpha b_n(t) C_n f_n^2, \tag{3}$$

where α is the effective capacitance of the computing chipset for each MD.

C. Model Parameter Transmission and Aggregation

After performing the local training, the MDs then upload their updated local model parameters back to the edge server. Let $\epsilon_n(t)$ denote the upload decision of the n-th MD at the iteration t, where $\epsilon_n(t) = 1$ means the n-th MD is assigned to a subchannel and is willing to upload parameters to the edge server through the assigned channel, and $\epsilon_n(t) = 0$ indicates that it is not assigned to a subchannel or keeps silent. Intuitively, a MD is more likely to upload if it has sufficient remaining energy while in a good channel state. For the uplink transmission, we adopt OFDMA technique, where the channels are orthogonal cross the different links. We assume that there are L orthogonal subchannels in the FL system and each MD can only occupy at most one subchannel. Let $h_n(t)$ denote the uplink channel gain between the n-th MD and the edge server in the iteration t, where the channel gains of all subchannels between the server and a single MD are same.

We model the channel gain $h_n(t)$ as a discrete-state block fading, where the channel gain between the n-th MD and edge server is a discrete random variable with a general distribution $\Pr[\bar{h}_n]$ [31], [35], [36], [37], [38]. We further assume that $h_n(t)$ stays invariant within each iteration and are IID across different iterations and MDs [31], [35], [36], [37], [38].

If the n-th MD is allowed to upload ($\epsilon_n(t) = 1$), it will transmit the local model parameters to the edge server with power $P_n(t)$ ($P_n(t) > 0$) in the uplink. Otherwise, the n-th MD keeps silent ($P_n(t) = 0$). We assume that the size of the local training parameters of all MDs is the same, which is denoted by M.[4] The uplink transmission rate for the n-th MD is given by

$$R_n(t) = \epsilon_n(t) W \log_2\left(1 + \frac{P_n(t)h_n(t)}{\sigma^2}\right), \tag{4}$$

where W is the bandwidth of subchannel between each MD and the server, and σ^2 is the power of the additive white Gaussian noise. Moreover, the corresponding uplink transmission time is expressed as

$$\tau_n^{\mathrm{com}}(t) = \frac{\epsilon_n(t)M}{R_n(t)}. \tag{5}$$

The energy consumption of parameter uploading for the n-th MD can be expressed as

$$E_n^{\mathrm{com}}(t) = P_n(t)\tau_n^{\mathrm{com}}(t) = \frac{\epsilon_n(t)P_n(t)M}{R_n(t)}. \tag{6}$$

[4]We assume all MDs have the same structures of the local network model and bit precision (typically floating point precision) of the local network parameters, respectively.

Upon receiving the local updated parameters from the MDs, the edge server aggregates them into global parameters and then broadcasts them to all MDs through a downlink broadcast channel. Assume that the bandwidth of the broadcast channel is sufficiently wide and the transmit power of the edge server is much higher than that of the MDs. Therefore, we ignore the downlink transmission time without much loss of generality.

III. Constrained Markov Decision Process

In this section, we design and analyze the joint scheduling problem of computing and communication resources in the FL network. In the FL system, sequential decisions on local training and parameter transmission needs to be made for each iteration. From (1), we know that the remaining energy at the MD sides are correlated among adjacent iterations. We therefore formulate the joint computing and communication resource scheduling problem as a CMDP to maximize the long term system reward under energy and delay constraints. We assume that our dynamic model is not oriented to a single FL tasks by taking $T \to \infty$.

A. The Composition of CMDP

At the beginning of each iteration, each MD uploads its current local channel state and battery energy state to the edge server. Therefore, the edge server obtains global status information to take appropriate actions for all MDs. Once the decisions are made, the edge server will download the policy to each MD. Due to the extremely small size of data for state information and action decisions, we can ignore the delay and the energy for the transmission of the local states and the policy in the FL network. The CMDP formulation consists of the following components:

- *State:* We define the global state $\boldsymbol{S}(t)$ of the all MDs in the iteration t as $\boldsymbol{S}(t) = [\boldsymbol{h}(t), \boldsymbol{E}^{\mathrm{sta}}(t)]$, which is composed of the current global channel state $\boldsymbol{h}(t) = [h_1(t), \ldots, h_N(t)]$ and the current global remaining battery energy state $\boldsymbol{E}^{\mathrm{sta}}(t) = [E_1^{\mathrm{sta}}(t), \ldots, E_N^{\mathrm{sta}}(t)]$.
- *Action:* Let us denote the global action $\boldsymbol{A}(t)$ of all MDs in the iteration t by $\boldsymbol{A}(t) = [\boldsymbol{b}(t), \boldsymbol{\epsilon}(t), \boldsymbol{P}(t)]$, which consists of the number of bits of training data $\boldsymbol{b}(t) = [b_1(t), \ldots, b_N(t)]$, the upload decision $\boldsymbol{\epsilon}(t) = [\epsilon_1(t), \ldots, \epsilon_N(t)]$ and the transmit power $\boldsymbol{P}(t) = [P_1(t), \ldots, P_N(t)]$.
- *Transition probability:* According to the dynamic energy queue given in (1), the global remaining energy $\boldsymbol{E}^{\mathrm{sta}}(t)$ under action $\boldsymbol{A}(t)$ is a controlled Markov chain with the transition probability of

$$\begin{aligned}&\Pr\left[\boldsymbol{E}^{\mathrm{sta}}(t+1)|\boldsymbol{E}^{\mathrm{sta}}(t), \boldsymbol{A}(t)\right] \\ &\quad = \prod_n \Pr\Big[E_n^{\mathrm{arr}}(t) = E_n^{\mathrm{sta}}(t+1) \\ &\qquad\qquad - \left[E_n^{\mathrm{sta}}(t) - \left\lceil E_n^{\mathrm{com}}(t) + E_n^{\mathrm{cop}}(t)\right\rceil\right]^+\Big]. \end{aligned} \tag{7}$$

Since the energy queue dynamic is affected by both the training energy and communication energy, it is controlled by the actions $\boldsymbol{A}(t) = [\boldsymbol{b}(t), \boldsymbol{\epsilon}(t), \boldsymbol{P}(t)]$. The global state transition probability is also Markovian,

which is given by

$$\begin{aligned}&\Pr[\boldsymbol{S}(t+1)|\boldsymbol{S}(t),\boldsymbol{A}(t))]\\&=\Pr[\boldsymbol{h}(t+1)|\boldsymbol{S}(t),\boldsymbol{A}(t))]\\&\qquad\times\ \Pr\big[\boldsymbol{E}^{\mathrm{sta}}(t+1)|\boldsymbol{S}(t),\boldsymbol{A}(t))\big]\\&=\Pr[\boldsymbol{h}(t+1)]\Pr\big[\boldsymbol{E}^{\mathrm{sta}}(t+1)|\boldsymbol{S}(t),\boldsymbol{A}(t))\big],\end{aligned}\tag{8}$$

where the second equation is due to the IID property of wireless channel state, that is, the channel state is not a controlled variable.

- *Reward:* The model accuracy of the FL is difficult to quantify, and does not promise a closed-form. In most circumstances, one observes that the accuracy of FL training increases with the total size of local training data at each MD [13], [39]. Hence, we define the reward of the n-th MD by the product of its local training data size and its upload decision, i.e., $\sum_{n=1}^{N} b_n(t)\epsilon_n(t)$. If the MD is unable to upload the training parameters ($\epsilon_n(t)=0$), then its reward in the current iteration is 0.

We assume that each training iteration is synchronized across the MDs with the duration of τ. The total time for training and transmission should not exceed the duration of a iteration, i.e.,

$$\tau_n^{\mathrm{com}}(t)+\tau_n^{\mathrm{cop}}(t)\le\tau.\tag{9}$$

Moreover, the energy used for local training and uploading should not exceed the remaining energy $E_n^{\mathrm{sta}}(t)$ at the beginning of the iteration t, which is described by the energy causality constraint of

$$\lceil E_n^{\mathrm{com}}(t)+E_n^{\mathrm{cop}}(t)\rceil\le E_n^{\mathrm{sta}}(t).\tag{10}$$

From (9) and (10), we see that there is a tradeoff between the computing and communication phases due to limited time and battery energy in each training iteration. According to (3), if the number of training data bits b_n increases, the computing energy consumption will increase, which leaves less energy for the communication phase. In the meanwhile, according to (2), by increasing the training bits number b_n, the local training time will increase, which leaves less time for communication. Due to the above tradeoff, each MD needs to allocate time and energy wisely between the computing and communication phases. For example, the probability that the total remaining energy $E_n^{\mathrm{sta}}(t)$ at the n-th MD equals 0 can not exceed the energy outage probability constraint $\mathrm{Pr}_n^{\mathrm{out}}$, i.e.,

$$\Pr\left[E_n^{\mathrm{sta}}(t)=0\right]\le\mathrm{Pr}_n^{\mathrm{out}}.\tag{11}$$

Here, $E_n^{\mathrm{sta}}(t)=0$ does not mean that the MD is completely powered off. We adopt a dedicated battery to support the energy harvesting circuit and the control signaling in each training iteration. The dedicated battery stores the energy that arrives at random in each iteration, and provides energy for information feedback, local training and parameter updates during the FL process. We thus assume that the MDs can exhaust its battery before the next recharge cycle [40]. Furthermore, we assume the subchannels occupied by the MDs in the current iteration cannot exceed the total number of subchannels L in the system, i.e.,

$$\sum_{n=1}^{N}\epsilon_n(t)\le L.\tag{12}$$

Due to the randomness of states in each iteration and the correlation of states across these iterations, the edge server needs to make sequential decisions on b_n, ϵ_n and P_n along the time horizon. Without much loss of generality, we formulate the problem as an infinite horizon CMDP, resulting in the stationary policies which do not change with time. The definition of stationary control policy is given as follows.

Definition 1 (Stationary Control Policy): A stationary control policy is a mapping $\boldsymbol{S}\to\boldsymbol{A}$ from the state space to the action space $\mathcal{S}$, which is given by $\boldsymbol{\Omega}(\boldsymbol{S})=\boldsymbol{A}\in\boldsymbol{\mathcal{A}}$, $\forall\boldsymbol{S}\in\boldsymbol{\mathcal{S}}$. Hence, we denote the control policy of the all MD by $\boldsymbol{\Omega}(S)=(\boldsymbol{b},\boldsymbol{\epsilon},\boldsymbol{P})$. Let $\boldsymbol{\Omega}$ be the stationary feasible control policy which should satisfy constraints (9), (10), (11) and (12).

B. Constrained Markov Decision Process Problem

The formulation of CMDP is given in *Problem 1*. The aim is to find the efficient control policy that maximizes the total long-term average utility of all MDs under the energy outage constraints, the transmission power constraints, the delay constraints, the energy causality constraints and the subchannel constraints.

Problem 1 (CMDP Problem):

$$\max_{\boldsymbol{\Omega}}\ \mathcal{U}(\boldsymbol{\Omega})=\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}^{\boldsymbol{\Omega}}\left[\sum_{n=1}^{N}b_n(t)\cdot\epsilon_n(t)\right]\tag{13}$$

$$\begin{aligned}\text{s.t.}\ \ &\Pr\left[E_n^{\mathrm{sta}}(t)=0\right]\le\mathrm{Pr}_n^{\mathrm{out}},&\text{(13a)}\\&0\le P_n(t)\le P_n^{\max},&\text{(13b)}\\&\epsilon_n(t)\in\{0,1\},\quad(9),\ (10)\ \text{and}\ (12),\quad\forall n,\end{aligned}$$

where the expectation $\mathbb{E}^{\boldsymbol{\Omega}}[\cdot]$ is taken with respect to the steady-state distribution induced by the control policy $\boldsymbol{\Omega}$, and $P_n^{\max}$ is the maximum transmission power of the n-th MD. Besides, the constraint (13a) can be redescribed as a long-term description of the energy outage probability constraint, i.e.,

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}^{\boldsymbol{\Omega}}\big[\mathbf{1}[E_n^{\mathrm{sta}}(t)=0]\big]\le\mathrm{Pr}_n^{\mathrm{out}},\tag{14}$$

where $\mathbf{1}[\,\cdot\,]$ is an indicator function that takes on a value of 1 when the battery energy is exhausted at the n-th MD. The objective function (13) in *Problem 1* is the long-term average total utility of all MDs under the control policy $\boldsymbol{\Omega}$. In the following analysis, we will use $\boldsymbol{\mathcal{D}}(t)$ to denote the feasible region of all short-term constraints in *Problem 1* for convenience of expression. In order to deal with the long-term constraints (13a) in *Problem 1*, we use the Lagrangian duality method to take the constraints of the CMDP problem into account by augmenting the objective function with a weighted sum of the constraint functions, which is given by

$$L(\boldsymbol{\Omega},\boldsymbol{\gamma})=\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}^{\boldsymbol{\Omega}}[g(\boldsymbol{S}(t),\boldsymbol{\Omega},\boldsymbol{\gamma})],\tag{15}$$

$$\begin{aligned}&g(\boldsymbol{S}(t),\boldsymbol{\Omega},\boldsymbol{\gamma})\\&=\sum_{n=1}^{N}\big(b_n(t)\epsilon_n(t)-\gamma_n\mathbf{1}[E_n^{\mathrm{sta}}(t)=0]\big)+\gamma_n\,\mathrm{Pr}_n^{\mathrm{out}}.\end{aligned}\tag{16}$$

The corresponding Lagrange dual function $G(\boldsymbol{\gamma})$ is given by

$$G(\boldsymbol{\gamma}) = \max_{\boldsymbol{\Omega}} \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}^{\boldsymbol{\Omega}}[g(\boldsymbol{S}(t), \boldsymbol{\Omega}, \boldsymbol{\gamma})]. \quad (17)$$

There exists Lagrange multipliers $\boldsymbol{\gamma} \succeq 0$ such that $\boldsymbol{\Omega}^*$ maximizes the Lagrange function $L(\boldsymbol{\Omega}, \boldsymbol{\gamma})$. And we can get the following problem.

Problem 2 (Lagrange Dual Problem):

$$L^* = \min_{\boldsymbol{\gamma}} \max_{\boldsymbol{\Omega}} L(\boldsymbol{\Omega}, \boldsymbol{\gamma}) \quad (18)$$
$$\text{s.t.} \quad \boldsymbol{\gamma} \succeq 0, \quad \boldsymbol{\Omega} \in \boldsymbol{\mathcal{D}}(t), \forall t.$$

According to [41], there exists an optimal control policy $\boldsymbol{\Omega}^*$ and a series of non-negative Lagrangian multipliers $\boldsymbol{\gamma}^*$ such that $\boldsymbol{\Omega}^*$ maximizes the Lagrange function $L(\boldsymbol{\Omega}^*, \boldsymbol{\gamma}^*)$, and the inequality condition holds:

$$L(\boldsymbol{\Omega}, \boldsymbol{\gamma}^*) \le L(\boldsymbol{\Omega}^*, \boldsymbol{\gamma}^*) \le L(\boldsymbol{\Omega}^*, \boldsymbol{\gamma}), \ \forall \boldsymbol{\Omega}, \quad \forall \boldsymbol{\gamma} \succeq 0, \quad (19)$$

where $\boldsymbol{\Omega}^*$ and $\boldsymbol{\gamma}^*$ are the original optimal solution and the dual optimal solution, respectively. *Problem 1* can be regarded as an infinite-dimensional linear programming problem with the feasible region $\boldsymbol{\mathcal{D}}(t)$, which is a special type of convex problem. Thus, the duality gap between the original optimal and the dual optimal is 0. The original optimal solution is obtained by solving the dual problem.

Generally, Bellman equation is a necessary condition for a dynamic programming to be optimized. Given Lagrange multipliers $\boldsymbol{\gamma}$, the classical infinite-horizon CMDP problem *Problem 1* can be solved by the Bellman equation [36]. Thus, we can obtain the following equation,

$$G(\boldsymbol{\gamma}) + V(\boldsymbol{S}(t)) = \max_{\boldsymbol{\Omega}} \Bigg\{ g(\boldsymbol{S}(t), \boldsymbol{\Omega}, \boldsymbol{\gamma}) + \sum_{\boldsymbol{S}(t+1)} \Pr(\boldsymbol{S}(t+1)|\boldsymbol{S}(t), \boldsymbol{\Omega})\, V(\boldsymbol{S}(t+1)) \Bigg\}, \quad \forall \boldsymbol{S}(t), \quad (20)$$

where $V(\boldsymbol{S}(t))$ is the value function representing the average utility obtained by the control policy $\boldsymbol{\Omega}$ from each global state $[\boldsymbol{h}(t), \boldsymbol{E}^{\text{sta}}(t)]$. According to (8), we know that the channel states possess independent statistical characteristics, which is not affected by the control policy. We can further simplify the Bellman equation by taking the expectation of (20) on the global channel state $\boldsymbol{h}(t)$.

Lemma 1 (Equivalent Bellman Equation): Given a series of Lagrange multipliers $\boldsymbol{\gamma}$, the objective function (15) can be solved by the equivalent Bellman equation as follows:

$$G(\boldsymbol{\gamma}) + V(\boldsymbol{E}^{\text{sta}}(t)) = \max_{\boldsymbol{\Omega}(\boldsymbol{E}^{\text{sta}})} \Big\{ \overline{g}\big(\boldsymbol{E}^{\text{sta}}(t), \boldsymbol{\Omega}(\boldsymbol{E}^{\text{sta}})\big) + \sum_{\boldsymbol{E}^{\text{sta}}(t+1)} \Pr\big(\boldsymbol{E}^{\text{sta}}(t+1)|\boldsymbol{E}^{\text{sta}}(t), \times \boldsymbol{\Omega}(\boldsymbol{E}^{\text{sta}})\big)\, V(\boldsymbol{E}^{\text{sta}}(t+1)) \Big\} \quad \forall \boldsymbol{E}^{\text{sta}}(t), \quad t > 0, \quad (21)$$

where the expectation of the value function $V(\boldsymbol{h}(t), \boldsymbol{E}^{\text{sta}}(t))$ is

$$V(\boldsymbol{E}^{\text{sta}}(t)) = \mathbb{E}_{\boldsymbol{h}(t)}[V(\boldsymbol{h}(t), \boldsymbol{E}^{\text{sta}}(t))]. \quad (22)$$

Similarly, by taking the expectation over the channel state, we have

$$\overline{g}\big(\boldsymbol{E}^{\text{sta}}(t), \boldsymbol{\Omega}(\boldsymbol{E}^{\text{sta}})\big) = \mathbb{E}_{\boldsymbol{h}(t)}\big[g(\boldsymbol{h}(t), \boldsymbol{E}^{\text{sta}}(t), \boldsymbol{\Omega}, \boldsymbol{\gamma})\big], \quad (23)$$

$$\Pr(\boldsymbol{E}^{\text{sta}}(t+1)|\boldsymbol{E}^{\text{sta}}(t), \boldsymbol{\Omega}(\boldsymbol{E}^{\text{sta}})) = \mathbb{E}_{\boldsymbol{h}(t)}\big[\Pr(\boldsymbol{E}^{\text{sta}}(t+1)|\boldsymbol{h}(t), \boldsymbol{E}^{\text{sta}}(t), \boldsymbol{\Omega}(\boldsymbol{E}^{\text{sta}})\big]. \quad (24)$$

Moreover, $\boldsymbol{\Omega}(\boldsymbol{E}^{\text{sta}}) = \{\boldsymbol{\Omega}(\boldsymbol{h}(t), \boldsymbol{E}^{\text{sta}}(t))|\forall \boldsymbol{h}(t)\}$ is a policy set under a given global energy state $\boldsymbol{E}^{\text{sta}}(t)$ for all possible channel states.

From the equivalent Bellman equation (21), we notice that the equation is composed by a series of linear equations, where the dimensions of these equations depend on the number of value functions $V(\boldsymbol{E}^{\text{sta}}(t))$. Hence, for any global energy state $\boldsymbol{E}^{\text{sta}}(t)$ and the global channel state $\boldsymbol{h}(t)$, the optimized control policy $\boldsymbol{\Omega}^*$ in (15) can be obtained by maximizing the right-hand side of (21).

IV. Approximate Markov Decision Process and Stochastic Learning

In this section, we use approximate MDP and stochastic learning methods to analyze and simplify the resource scheduling problem, and design online algorithms to obtain the resource scheduling policy for the FL system.

A. Approximate Markov Decision Process

According to (21), the global energy state value function $V(\boldsymbol{E}^{\text{sta}}(t))$ is unknown, which holds a great difficulty for solving the control policy in the FL system. Due to the existence of the huge state-action space, we are unable to get the value function with the conventional value iteration method. However, we can obtain the value function and develop a solution of *Problem 1* by the value approximation and online stochastic learning. Assumed that we have obtained the value function $V(\boldsymbol{E}^{\text{sta}}(t))$ through value approximation and online stochastic learning. Thus, the MDP problem can be solved as follow.

Problem 3: For a given value function $V(\boldsymbol{E}^{\text{sta}}(t))$, find the optimized action $\boldsymbol{\Omega}^*(\boldsymbol{E}^{\text{sta}}(t))$, which is satisfied to the Equivalent Bellman's equation in (21). The optimized control policy can be rewritten as

$$\begin{aligned} &\boldsymbol{\Omega}^*(\boldsymbol{E}^{\text{sta}}(t)) \\ &= \arg \max_{\boldsymbol{\Omega}(\boldsymbol{E}^{\text{sta}}(t))} \mathbb{E}_{\boldsymbol{h}(t)}\big\{ g(\boldsymbol{S}(t), \boldsymbol{\Omega}(\boldsymbol{E}^{\text{sta}}(t)), \boldsymbol{\gamma}) \\ &\quad + \sum_{\boldsymbol{E}^{\text{sta}}(t+1)} \Pr\big(\boldsymbol{E}^{\text{sta}}(t+1)|\boldsymbol{h}(t), \boldsymbol{E}^{\text{sta}}(t), \\ &\quad \boldsymbol{\Omega}(\boldsymbol{E}^{\text{sta}}(t))\big)\, V(\boldsymbol{E}^{\text{sta}}(t+1))\big\}, \\ &\text{s.t.} \quad 0 \le P_n(t) \le P_n^{\max}, \\ &\epsilon_n(t) \in \{0, 1\}, \\ &\quad (9),\ (10)\ \text{and}\ (12), \quad \forall n. \end{aligned} \quad (25)$$

Given the global state value function $V(\boldsymbol{E}^{\text{sta}}(t))$ and the realization of the global channel state $\boldsymbol{h}$, the *Problem 3* then becomes a static optimization problem.

B. Stochastic Learning

By the feature-based method, the energy state value function $V(\boldsymbol{E}^{\text{sta}})$ can be approximated by a linear form of the state value function of the n-th MD $V_n(E_n^{\text{sta}})$. The global energy state value function $V(\boldsymbol{E}^{\text{sta}})$ and a series of Lagrange multipliers $\boldsymbol{\gamma}$ will be updated according to the current energy state and channel state information. Then, the proposed linear approximation architecture for the global energy state value function $V(E_n^{\text{sta}})$ is given by

$$V(\boldsymbol{E}^{\text{sta}}) = V\big(E_1^{\text{sta}}, \ldots, E_n^{\text{sta}}, \ldots, E_N^{\text{sta}}\big) \approx \sum_{n=1}^{N} \sum_{l \in \boldsymbol{Q}_n} V_n(l) \boldsymbol{I}\big[E_n^{\text{sta}} = l\big] = \boldsymbol{W}^T \boldsymbol{F}(\boldsymbol{E}^{\text{sta}}), \tag{26}$$

where $\boldsymbol{Q}_n$ means energy state set of the n-th MD, that is $\boldsymbol{Q}_n = \{0, 1, 2, \ldots, E_n^{\max}\}$. The parameter vector $\boldsymbol{W}$ and the feature $\boldsymbol{F}(\boldsymbol{E}^{\text{sta}})$ can be elaborated as

$$\boldsymbol{W} = \big[V_1(0), \ldots, V_1(E_1^{\max}), \ldots, V_N(0), \ldots, V_N(E_N^{\max})\big]^T, \tag{27}$$

$$\boldsymbol{F}(\boldsymbol{E}^{\text{sta}}) = \Big[\boldsymbol{I}[E_1^{\text{sta}} = 0], \ldots, \boldsymbol{I}[E_1^{\text{sta}} = E_1^{\max}], \cdots \boldsymbol{I}[E_N^{\text{sta}} = 0], \ldots, \boldsymbol{I}[E_N^{\text{sta}} = E_N^{\max}]\Big]^T. \tag{28}$$

Thus, we can calculate the global energy state value function by the linear form of all MDs. The value function of global energy state $\boldsymbol{E}^{\text{sta}} = \{E_1^{\text{sta}}, \ldots, E_N^{\text{sta}}\}$ can be expressed as

$$V\Big(\boldsymbol{E}^{\text{sta}}\Big) \approx \sum_{n=1}^{N} V_n\big(E_n^{\text{sta}}\big), \quad E_n^{\text{sta}} \in \boldsymbol{Q}_n. \tag{29}$$

The global energy state value function $V(\boldsymbol{E}^{\text{sta}})$ is the same as the cardinality of the global energy state $\boldsymbol{E} = [E_1, \ldots, E_N]$, and its number is $\prod_{n=1}^{N}(E_n^{\max} + 1)$. However, the number of the linear approximation energy state value function of all MDs is $\sum_{n=1}^{N}(E_n^{\max} + 1)$. Through linear approximation architecture, we exploit the state value function of each MD $V_n(E_n^{\text{sta}})$ with a small state space to represent the global energy state value function $V(\boldsymbol{E}^{\text{sta}})$ with huge state space.

According to *Lemma* 1 and the linear approximation, we can obtain the following equations, i.e.,

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{h}}\Bigg\{\sum_{\boldsymbol{E}^{\text{sta}}(t+1)} \Pr\big(\boldsymbol{E}^{\text{sta}}(t+1)|\boldsymbol{h}(t), \boldsymbol{E}^{\text{sta}}(t), \boldsymbol{\Omega}(\boldsymbol{h}(t), \boldsymbol{E}^{\text{sta}}(t))\big) V(\boldsymbol{E}^{\text{sta}}(t+1))\Bigg\} \\
&= \mathbb{E}_{\boldsymbol{h}}\Bigg\{\sum_{\boldsymbol{E}^{\text{sta}}(t+1)} \Bigg(\prod_{n=1}^{N} \Pr\big(E_n^{\text{sta}}|\boldsymbol{h}(t), \boldsymbol{E}^{\text{sta}}(t), \boldsymbol{\Omega}(\boldsymbol{h}(t), \boldsymbol{E}^{\text{sta}}(t))\big) \sum_{n=1}^{N} V_n(E_n^{\text{sta}}(t+1))\Bigg)\Bigg\} \\
&= \mathbb{E}_{\boldsymbol{h}}\Bigg\{\sum_{n=1}^{N} \sum_{E_n^{\text{sta}}(t+1) \in \boldsymbol{Q}_n} \Pr\big(E_n^{\text{sta}}(t+1)|\boldsymbol{h}(t), \boldsymbol{E}^{\text{sta}}(t), \boldsymbol{\Omega}(\boldsymbol{h}(t), \boldsymbol{E}^{\text{sta}}(t))\big) V_n(E_n^{\text{sta}}(t+1))\Bigg\} \\
&= \mathbb{E}_{\boldsymbol{h}}\Bigg\{\sum_{n=1}^{N} \sum_{A_n(t)} \Pr(A_n(t)) V_n(E_n^{\text{sta}}(A_n(t), \Omega_n(\boldsymbol{S}(t))))\Bigg\},
\end{aligned} \tag{30}$$

where the post-action energy state of the n-th MD $E_n^{\text{sta}}(A_n(t), \Omega_n(\boldsymbol{S}(t)))$ can be defined as

$$E_n^{\text{sta}}(A_n(t), \Omega_n(\boldsymbol{S}(t))) = \min\Big\{\Big[E_n^{\text{sta}}(t) - \big\lceil E_n^{\text{com}}(t) + E_n^{\text{cop}}(t)\big\rceil\Big]^+ + A_n(t), E_n^{\max}\Big\}. \tag{31}$$

The equation (30) holds due to the state transition probability in (7) and the state update in (31). Thus, we can get the following optimized policy by (30),

$$\boldsymbol{\Omega}^*(\boldsymbol{E}(t)) = \arg \max_{\boldsymbol{\Omega}(\boldsymbol{E}(t))} \mathbb{E}_{\boldsymbol{h}}\Bigg\{g(\boldsymbol{S}(t), \boldsymbol{\Omega}(\boldsymbol{S}(t)), \boldsymbol{\gamma}) + \sum_{n=1}^{N} \sum_{A_n(t)} \Pr(A_n(t)) V_n(E_n^{\text{sta}}(A_n(t), \Omega_n(\boldsymbol{S}(t))))\Bigg\}. \tag{32}$$

According to the linear value approximation structure (29) and (32), the control policy problem can be re-written as the following problem.

Problem 4 (Approximate Control Policy Problem):

$$\begin{aligned}
\max_{\boldsymbol{\Omega}^*} \ & \mathbb{E}_{\boldsymbol{h}}\Bigg\{g(\boldsymbol{S}(t), \boldsymbol{\Omega}(\boldsymbol{S}(t)), \boldsymbol{\gamma}) + \sum_{n=1}^{N} \sum_{A_n(t)} \Pr(A_n(t)) V_n(E_n^{\text{sta}}(A_n(t), \Omega_n(\boldsymbol{S}(t))))\Bigg\}, \\
\text{s.t.} \ & 0 \le P_n(t) \le P_n^{\max}, \\
& \epsilon_n(t) \in \{0, 1\}, \\
& (9),\ (10)\ \textit{and}\ (12), \forall n.
\end{aligned} \tag{33}$$

Since $E_n^{\text{sta}}(A_n(t), \Omega_n(\boldsymbol{S}(t)))$ represents the update of the local energy state, we need to calculate the objective function of (33) for each local energy state, and derive the objective function over all local energy states. To solve (33), we expand $V(E_n^{\text{sta}}(A_n(t), \Omega_n(\boldsymbol{S}(t))))$ in (33) using Taylor expansion as follows [42], [43]:

$$V(E_n^{\text{sta}}(A_n(t), \Omega_n(\boldsymbol{S}(t)))) = V(E_n^{\text{sta}}(t)) + (A_n(t) - \big\lceil E_n^{\text{com}}(t) + E_n^{\text{cop}}(t)\big\rceil) V'(E_n^{\text{sta}}(t)), \tag{34}$$

where

$$V'(E_n^{\text{sta}}(t)) = \big[V(E_n^{\text{sta}}(t) + 1) - V(E_n^{\text{sta}}(t) - 1)\big]/2. \tag{35}$$

The optimization objective in (33) can be expressed as follow,

$$\begin{aligned}
\max_{\boldsymbol{\Omega}} \ & \mathbb{E}_{\boldsymbol{h}}\{g(\boldsymbol{S}(t), \boldsymbol{\Omega}(\boldsymbol{S}(t), \boldsymbol{\gamma}) \\
& + \sum_{n=1}^{N} \sum_{A_n(t)} \Pr(A_n(t))\big(V(E_n^{\text{sta}}(t)) + (A_n(t) \\
& - \big\lceil E_n^{\text{com}}(t) + E_n^{\text{cop}}(t)\big\rceil) V'(E_n^{\text{sta}}(t))\big)\},
\end{aligned} \tag{36}$$

where (36) is a static mixed variable optimization problem, in which $\boldsymbol{b}$ and $\boldsymbol{P}$ are continuous variables, while $\boldsymbol{\epsilon}$ are discrete

variables. Besides, the ceiling operator $\lceil\cdot\rceil$ is difficult to handle, which brings great difficulties to the optimization problem. In order to solve the problem caused by the ceiling operator $\lceil\cdot\rceil$, we introduce a series of auxiliary variables $\Delta E_n(t), \forall n$ to simplify the optimization problem. The optimization problem can be further described as follows:

$$\begin{aligned}\max_{\boldsymbol{b},\boldsymbol{\epsilon},\boldsymbol{P}}\ & g(\boldsymbol{S}(t),\boldsymbol{\Omega}(t),\boldsymbol{\gamma}) + \sum_{n=1}^{N}\sum_{A_n(t)} \Pr(A_n(t)) \\ & \times (A_n(t) - \Delta E_n(t))\, V'(E_n(t)) \\ \text{s.t.}\ & (9)\ \text{and}\ (12), \\ & E_n^{\text{com}}(t) + E_n^{\text{cop}}(t) - \Delta E_n(t) \le 0, \\ & \Delta E_n(t) \in \{0,1,2,\ldots,E_n(t)\}, \\ & \epsilon_n(t) \in \{0,1\}, \\ & 0 \le P_n(t) \le P_n^{\max}, \quad \forall n, \end{aligned} \tag{37}$$

where the auxiliary variable is

$$\begin{aligned}\Delta E_n(t) &= \lceil E_n^{\text{com}}(t) + E_n^{\text{cop}}(t)\rceil \\ &= \left\lceil \alpha b_n(t) C_n f_n^2 + \frac{\epsilon_n(t) P_n(t) d}{R_{n,s}(t)} \right\rceil. \end{aligned} \tag{38}$$

Note that the constraints (12) describes the subchannel constraints of all MDs, we ignore the constraints for the time being to simplify the optimization problem. Given a typical MD n, we can obtain the following optimization problem by further analysis and simplification of (36), i.e.,

$$\begin{aligned}\max_{b_n,\epsilon_n,P_n}\ & \frac{\Delta E_n(t) - E_n^{\text{com}}(t)}{\alpha C_n f_n^2}\epsilon_n(t) - \Delta E_n(t)\, V'(E_n(t)), \\ \text{s.t.}\ & T_n^{\text{com}}(t) + T_n^{\text{cop}}(t) \le \tau, \\ & \Delta E_n(t) \in \{0,1,2,\ldots,E_n(t)\}, \\ & \epsilon_n(t) \in \{0,1\}, \\ & 0 \le P_n(t) \le P_n^{\max}, \quad \forall n. \end{aligned} \tag{39}$$

From (39), we can draw the following conclusion obviously: if $\Delta E_n(t) \le E_n^{\text{th}}(t)$, then $P_n(t) = 0$ and $\epsilon_n(t) = 0$, in which $E_n^{\text{th}}(t)$ means the threshold energy of the n-th MD at the current iteration, and it can be expressed as

$$E_n^{\text{th}}(t) = \frac{\tau\sigma^2}{h_n(t)}\left(2^{\frac{M}{W\tau}} - 1\right). \tag{40}$$

When $\Delta E_n(t) \le E_n^{\text{th}}$, the energy consumed by the n-th MD at the current iteration is insufficient to support uploading model parameters to the edge server within the iteration duration τ.

Due to the first constraint of (39), we obtain the upper bound of energy consumption by the n-th MD, that is,

$$\Delta E_n(t) \le (\tau - T_n^{\text{com}}(t))\alpha f_n^3 + T_n^{\text{com}}(t) P_n(t). \tag{41}$$

When $\frac{1}{\alpha C_n f_n^2} - V'(E_n(t)) \le 0$, the energy consumption $\Delta E_n(t)$ is 0 obviously. Correspondingly, the transmission power $P_n(t)$ of the n-th MD, the transmission decision $\epsilon_n(t)$ of the n-th MD and the batch size of local training data $b_n(t)$ are all 0. When $\frac{1}{\alpha C_n f_n^2} - V'(E_n(t)) > 0$ and $\Delta E_n(t) > E_n^{\text{th}}(t)$, since the value of $P_n(t)$ is related to the value of $\Delta E_n(t)$ and $\Delta E_n(t) \le (\tau - T_n^{\text{com}}(t))\alpha f_n^3 + T_n^{\text{com}}(t) P_n(t)$, the energy consumption $\Delta E_n(t)$ of the n-th MD takes the maximum value, i.e.,

$$\Delta E_n(t) = \left(\tau - \frac{M}{R_{n,s}(t)}\right)\alpha f_n^3 + \frac{M}{R_{n,s}(t)} P_n(t). \tag{42}$$

Since $P_n(t) \in (0, P_n^{\max}]$ and $\Delta E_n(t)$ increases monotonically as $P_n(t)$ increases, there is a maximum value of $\Delta E_n^{\max}(t)$ as a function of $P_n(t)$, which can be expressed as

$$\Delta E_n^{\max}(t) = \left(\tau - \frac{M}{R_{n,s}^{\max}(t)}\right)\alpha f_n^3 + \frac{M}{R_{n,s}^{\max}(t)} P_n^{\max}. \tag{43}$$

Thus, when $\Delta E_n(t) > \Delta E_n^{\max}(t)$, the transmission power $P_n(t)$ is $P_n^{\max}$, $\epsilon_n(t) = 1$ and the batch size $b_n(t)$ for local training can be expressed as

$$b_n(t) = \frac{\Delta E_n^{\max}(t) - E_{n,\max}^{\text{com}}(t)}{\alpha C_n f_n^2}, \tag{44}$$

$$E_{n,\max}^{\text{com}}(t) = \frac{M P_n^{\max}}{W \log_2\left(1 + \frac{P_n^{\max} h_n(t)}{\sigma^2}\right)}. \tag{45}$$

Then for $E_n^{\text{th}}(t) < \Delta E_n(t) \le \Delta E_n^{\max}(t)$, according to the relationship between $\Delta E_n(t)$ and $P_n(t)$, we can express $P_n(t)$ by $\Delta E_n(t)$, which is given by

$$\begin{aligned} P_n(t) &= \frac{\text{lambertW}\left(\frac{B_n(t)}{Z_n(t)} e^{\frac{C_n(t)}{Z_n(t)}}\right)}{\frac{B_n(t)}{Z_n(t)} h_n(t)} - \frac{\sigma^2}{h_n(t)}, \\ B_n(t) &= -\frac{M}{h_n(t)}, \\ C_n(t) &= \frac{W(\Delta E_n(t) - \alpha f_n^3 \tau)}{\ln 2} \ln \sigma^2 - \frac{M\sigma^2}{h_n(t)} - \alpha f_n^3 M, \\ Z_n(t) &= \frac{W(\Delta E_n(t) - \alpha f_n^3 \tau)}{\ln 2}. \end{aligned} \tag{46}$$

And $\epsilon_n(t) = 1$, lambertW means Lambert W Function, which is the inverse function of $f(w) = w \cdot \exp(w)$. And it is a special function that cannot be represented by an expression. From this, we get the objective function of the variable of ΔE_n by substituting $P_n(t)$ into the objective function in (39),

$$\begin{aligned} F_n(t) = \max_{\Delta E_n(t)}\ & \frac{1}{\alpha C_n f_n^2}\Big(\min\{\Delta E_n(t), \Delta E_n^{\max}(t)\} \\ & - \frac{dP_n(\Delta E_n(t))}{R_{n,s}(\Delta E_n(t))}\Big)\epsilon_n(t) - \Delta E_n(V_n'(E_n(t)) \\ \text{s.t.}\ & \Delta E_n(t) \in \left\{0,1,\ldots,\min\left\{E_n^{\text{sta}}(t), \lceil \Delta E_n^{\max}(t)\rceil\right\}\right\}. \end{aligned} \tag{47}$$

By searching in the $\{0,1,\ldots,\min\{E_n(t), \lceil E_n^{\max}(t)\rceil\}\}$, we can find the optimized energy consumption value $\Delta \hat{E}_n(t)$ of the n-th MD for the maximum value of the objective function in (47). For convenience, we assume 0/0 = 0 for the term of $\frac{dP_n(\Delta E_n(t))}{R_{n,s}(\Delta E_n(t))}$ in this paper. Recalling the subchannel constraint (12) that we ignored earlier, we now analyze it. According to (47), we get the optimized objective function $F_n(t)$ of each MD. According to (12), we know that wireless communication resources in our system are limited. To ensure the performance of FL, MDs with low training

Algorithm 1: The Pseudocode of the Proposed Static Mixed Variable Optimization Problem

Input: input $\boldsymbol{E}^{\mathbf{sta}}(t), \boldsymbol{h}(t), f_n, P_n^{\max}, V_n'(t), \forall n$;
Output: output result $\boldsymbol{P}^*(t), \boldsymbol{b}^*(t), \boldsymbol{\epsilon}^*(t)$;
for *The n-th MD,* $n \in 1, \ldots, N$ **do**
  Calculate threshold energy $E_n^{\mathrm{th}}(t)$ and the maximum energy consumption $\Delta E_n^{\max}(t)$ for computation and communication in the t-th iterartion;
  for $\Delta E_n(t) \in \{0, 1, \ldots, \min\{E_n(t), \lceil \Delta E_n^{\max}(t) \rceil\}\}$ **do**
    if $\Delta E_n(t) \leq E_n^{\mathrm{th}}(t)$ **then**
      $P_n(t) = 0, b_n(t) = 0, \epsilon_n(t) = 0$;
    else if $E_n^{\mathrm{th}}(t) < \Delta E_n(t) \leq \Delta E_n^{\max}(t)$ **then**
      The solution of $P_n(t)$ and $b_n(t)$ can refer to (46) and $\epsilon_n(t) = 1$;
    else
      $P_n(t) = P_n^{\max}$, the solution of $b_n(t)$ can refer to (44), $\epsilon_n(t) = 1$;
    end if
    Substituting the values of $P_n(t)$ and $\epsilon_n(t)$ into the objective function (39) ;
  By searching in the $\{0, 1, \ldots, \min\{E_n(t), \lceil \Delta E_n^{\max}(t) \rceil\}\}$, the optimized energy consumption value $\Delta \hat{E}_n(t)$ of the n-th MD for the maximum value of the objective function in (47) can be found, and $\hat{P}_n(t), \hat{b}_n(t), \hat{\epsilon}_n(t)$ can be calculated.
if $\| \hat{\boldsymbol{\epsilon}} \|_1 \leq L$ **then**
  The global optimized solution is equal to the solution obtained by the respective MD, i.e., $\boldsymbol{P}^*(t) = \hat{\boldsymbol{P}}(t)$, $\boldsymbol{b}^*(t) = \hat{\boldsymbol{b}}(t)$.
else
  The edge server will select L MDs with the largest $F_n(t)$ for FL training. If the n-th MD is selected by the server, then $P_n^*(t) = \hat{P}_n(t), \epsilon_n^*(t) = \hat{\epsilon}_n(t)$ and $b_n^*(t) = \hat{b}_n(t)$, otherwise $P_n^*(t) = 0, \epsilon_n^*(t) = 0$ and $b_n^*(t) = 0$.
end if

performance are not scheduled to upload their local models for global model aggregation, whereas the limited bandwidth can be assigned to mobile devices with superior local training performance. Specifically, if $\|\hat{\boldsymbol{\epsilon}}\|_1 \leq L$, the global optimized solution is equal to the solution obtained by the respective MD, i.e., $P_n^*(t) = \hat{P}_n(t), b_n^*(t) = \hat{b}_n(t)$. Then, when $\|\hat{\boldsymbol{\epsilon}}\|_1 > L$, the edge server will select L MDs with the largest $F_n(t)$ for FL training. If the n-th MD is selected by the edge server to upload parameters, then $P_n^*(t) = \hat{P}_n(t), \epsilon_n^*(t) = \hat{\epsilon}_n(t)$ and $b_n^*(t) = \hat{b}_n(t)$, otherwise $P_n^*(t) = 0, \epsilon_n^*(t) = 0$ and $b_n^*(t) = 0$. Algorithm 1 reports the pseudocode of the proposed static mixed variable optimization problem.

In the previous section, we assumed that the state value function $V(\boldsymbol{E}^{\mathrm{sta}})$ has been given. However, we need to know the state value function of each MD accurately so that we can make efficient control decisions. We utilize stochastic learning

Algorithm 2: The Specific Flow of the Stochastic Learning

1 Initialize the respective energy state value function vectors $\boldsymbol{V}^0$ and the Lagrange multipliers vectors $\boldsymbol{\gamma}^0$ of all MDs;
2 Based on the observed local states, a series of parameters and the local energy value functions $\boldsymbol{V}^t$ of each MD, the control action can be calculated by Algorithm 1 at the beginning of the iteration t;
3 Based on the observed local states, the control actions and the instantaneous rewards of the system, the energy state value function $\boldsymbol{V}^{t+1}$ and Lagrange multipliers vectors $\boldsymbol{\gamma}^{t+1}$ can be updated by (48), (49) and (50);
4 If $\| \boldsymbol{V}^{t+1} - \boldsymbol{V}^t \| < \delta_v$ and $\| \boldsymbol{\gamma}^{t+1} - \boldsymbol{\gamma}^t \| < \delta_\gamma$, stop; otherwise, set $t = t + 1$ and go back to step 2.

and distributed online algorithm to estimate the value function $V(\boldsymbol{E}^{\mathrm{sta}})$ and the Lagrange multipliers $\boldsymbol{\gamma}$ based on the current state. The updates of the value function V at the end of the iteration t can be given by (48) and (49).

$$V_n^{t+1}(l) = \begin{cases} (1 - \epsilon_v^t) V_n^t(l) + \epsilon_v^t \Delta V_n^{t+1}(l) & \text{if} \quad l = \boldsymbol{E}_n^{t+1} \\ V_n^t(l) & \text{if} \quad l \neq \boldsymbol{E}_n^{t+1}, \end{cases} \tag{48}$$

$$\Delta V_n^{t+1}(l) = b_n(t)\epsilon_n(t) - \gamma_n^t \mathbf{1}[l = 0] + \sum_{A_n} \Big\{ \Pr(A_n)(V_n^t(l(\Delta E_n^{t+1}, A_n)) - V_n^t(l(A_n))) \Big\}. \tag{49}$$

Moreover, the Lagrange multipliers updates at each MD are given by

$$\gamma_n^{t+1} = [\gamma_n^t + \epsilon_\gamma^t (\mathbf{1}[E_n^{t+1} = 0] - \Pr_n^{\mathrm{th}})]^+. \tag{50}$$

In the above equations, $(\{\epsilon_v^t\}, \{\epsilon_\gamma^t\})$ are the sequences of iteration size, which satisfy

$$\sum_{t=0}^{\infty} \epsilon_v^t = \infty, \ \epsilon_v^t > 0, \ \lim_{t \to \infty} \epsilon_v^t = 0,$$
$$\sum_{t=0}^{\infty} \epsilon_\gamma^t = \infty, \ \epsilon_v^\gamma > 0, \ \lim_{t \to \infty} \epsilon_v^\gamma = 0,$$
$$\sum_{t=0}^{\infty} \Big[(\epsilon_v^t)^2 + (\epsilon_\gamma^t)^2 \Big] < \infty, \ \text{and} \ \lim_{t \to \infty} \frac{\epsilon_\gamma^t}{\epsilon_v^t} = 0. \tag{51}$$

The specific process of stochastic learning can refer to Algorithm 2.

C. Convergence Analysis

We need to provide effective analysis for the convergence of the online stochastic learning algorithm, which is shown in Algorithm 2. From the previous section, we notice that there are two different step size sequences $\{\epsilon_v^t\}$ and $\{\epsilon_\gamma^t\}$ in the stochastic learning process, which are used for the update of state value functions of MDs and Lagrange multipliers, respectively. Since the update of the Lagrangian multipliers $\boldsymbol{\gamma}$ and the update of the value function $\boldsymbol{V}$ occur simultaneously and $\epsilon_\gamma^t = \boldsymbol{o}(\epsilon_v^t)$, we can obtain $\gamma^{t+1} - \gamma^t = \boldsymbol{o}(\epsilon_v^t)$. Therefore,

we consider that the Lagrangian multipliers does not change when the state value function is updated. Therefore, we assume that the Lagrangian multipliers $\boldsymbol{\gamma}^t$ keep static when the value functions of the mobile devices are updated in (48).

The relationship between the global value function vector $\boldsymbol{V}$ and the parameter vector $\boldsymbol{W}$ can be expressed as

$$\boldsymbol{V} = \boldsymbol{M}\boldsymbol{W} \qquad \text{and} \qquad \boldsymbol{W} = \boldsymbol{M}^{\dagger}\boldsymbol{V}, \tag{52}$$

in which $\boldsymbol{M} \in \mathbb{R}^{|I_S| \times \sum_{n=1}^{N}(E_n^{\max}+1)}$ with the kth row ($k = 1, 2, \ldots, |I_S|$) equals to $\boldsymbol{F}(\boldsymbol{E}^k)$, $\boldsymbol{E}^k$ is the kth global energy state and $|I_S|$ is the cardinality of the system state. In addition, $\boldsymbol{M}^{\dagger} \in \mathbb{R}^{\sum_{n=1}^{N}(E_n^{\max}+1) \times |I_S|}$ means the mapping matrix from $\boldsymbol{V}$ to $\boldsymbol{W}$, which is the inverse mapping of the first equation of (52). We then have the following convergence lemma on the local state value function for each MD in the stochastic learning.

Lemma 2: (Convergence of State Value Function of Each MD): The convergence performance of the state value function can be described as follows.

1) The update of the state value function vector converge almost surely for any given initial parameter vector $\boldsymbol{W}^0$ and Lagrange multipliers γ, which can be expressed as

$$\lim_{t\to\infty} \boldsymbol{W}^t(\gamma) = \boldsymbol{W}^{\infty}(\gamma). \tag{53}$$

2) The local steady-state value function vector $\boldsymbol{W}^{\infty}$ satisfies the vector form of the following steady equivalent Bellman equation,

$$\theta \boldsymbol{I} + \boldsymbol{W}^{\infty}(\gamma) = \boldsymbol{M}^{\dagger}\boldsymbol{T}(\gamma, \boldsymbol{M}\,\boldsymbol{W}^{\infty}(\gamma)), \tag{54}$$

where $\boldsymbol{I}$ is a $\sum_{n=1}^{N}(E_n^{\max}+1)\times 1$ vector whose elements are all equal to 1, $\boldsymbol{T}$ represents a function mapping, which can be defined as,

$$\boldsymbol{T}(\gamma, \boldsymbol{V}) = \max_{\Omega}\{\overline{\boldsymbol{g}}(\gamma, \Omega) + \boldsymbol{P}(\Omega)\,\boldsymbol{V}\}, \tag{55}$$

where $\overline{\boldsymbol{g}}(\gamma, \Omega)$ is a $\sum_{n=1}^{N}(E_n^{\max}+1)\times 1$ vector of function $\overline{g}(\boldsymbol{E}, \Omega(\boldsymbol{E}))$, which is defined in (21). $\boldsymbol{P}(\Omega)$ is the matrix form of transition probability $\Pr(\boldsymbol{E}^{t+1}|\boldsymbol{E}^t, \Omega)$ defined in (21).

Proof: Following [43], we briefly proof the Lemma. Since we consider the stochastic channels, where the channel gain varies across the interval, it is easy to see that each state will be updated comparably often in the asynchronous learning algorithm. Quoting the conclusion from [43], the convergence properties of the asynchronous and synchronous updates are the same. Therefore, we just consider the convergence of related synchronous version for simplicity. According to the definition of parameter vector $\boldsymbol{W}$ and the bounded per-MD value function V_n, it is clearly that the update on the per-MD value function vector is equivalent to the update on the parameter vector and to prove the convergence of the Lemma is equivalent to prove the convergence of update on the parameter vector $\boldsymbol{W}$. The detailed proof can refer to [43]. ■

Due to $\epsilon_{\gamma}^t = \boldsymbol{o}(\epsilon_v^t)$, the ratio of step sizes between state value function and Lagrange multipliers can be expressed as $\epsilon_{\gamma}^t/\epsilon_v^t \to 0$ during the Lagrange multipliers update in (50), and the updates of the local state value function are much faster than the Lagrange multipliers. Thus, the Lagrange multipliers can be consider as quasi-invariant during the update of the local state value functions of each MD, and the update of the Lagrange multipliers will trigger another update process of the local state value function of each MD. According to [44], we can obtain that $\lim_{t\to\infty} || V_n^t - V_n^{\infty}(\gamma^t)|| = 0$, in which $V_n^{\infty}(\gamma^t)$ means the converged local state value function of nth MD with Lagrange multipliers γ^t. Therefore, the update of the local state value function can be considered as almost constant during the Lagrange multipliers' update. Then, we need to prove the convergence lemma of the Lagrange multipliers.

Lemma 3 (Convergence of the Lagrange multipliers): The iteration on the Lagrange multipliers $\boldsymbol{\gamma}$ converges almost surely to the set of minimum of $G(\boldsymbol{\gamma})$ in (17). Supposing that the Lagrange multipliers converge to $\boldsymbol{\gamma}^*$, then $\boldsymbol{\gamma}^*$ satisfies the average energy outage constraint in (11).

Proof: Quoting to [45, Lemma 4.2], $-G(\boldsymbol{\gamma})$ is a concave and continuously differentiable except at finitely many points where both right and left derivatives exist. Thus, $G(\boldsymbol{\gamma})$ is a convex function of γ. Since the energy consumption policy of each MD is discrete, we can obtain that $\Omega^*(\gamma) = \Omega^*(\gamma + \Delta_{\gamma})$, i.e., $\nabla_{\gamma} = (\Omega^*(\gamma + \Delta_{\gamma}) - \Omega^*(\gamma))/\Delta_{\gamma} = 0$. Thus, $\partial G(\gamma^t)/\partial \gamma^t$ can be expressed as $\partial G(\gamma^t)/\partial \gamma^t = \mathbb{E}^{\Omega^*(\gamma^t)}\{\Pr_n^E - \mathbf{1}[E_n(t) = 0]\}$, where $\Omega^*(\gamma^t) = \arg\max_{\Omega} G(\gamma^t)$. By the standard stochastic approximation theorem [46], the dynamics of the Lagrange multipliers update can be represented by ordinary differential equation (ODE). According to [47], we know that the ODE equals to $\partial G(\gamma^t)/\partial \gamma^t$. Thus, the aforementioned ODE will converge to $\partial G(\gamma^t)/\partial \gamma^t = 0$, i.e., the average energy outage constraints are satisfied. ■

According to Lemmas 2 and 3, the iteration on local state value function and the Lagrange multipliers in Algorithm 2 will converge. Next, we analyze the complexity, optimality and implementation of the stochastic learning algorithm. The proposed algorithm has a simple structure with a computational complexity of $\mathcal{O}(\sum_{n=1}^{N}(E_n^{\max}+1))$, which grows linearly with the energy state space of all MDs. Since we obtained the value function by the value approximation and online stochastic learning, the proposed algorithm is a low-complexity suboptimal algorithm and the following simulations will verify the effectiveness of the proposed algorithm. Moreover, each MD only transmits its own energy state information to the server, and the server broadcasts the control policy to all MDs. Compared to the traditional FL model parameters, the signaling overhead of the proposed algorithm is quite small (even negligible). Hence, the proposed algorithm is easy to implement on the edge server.

Remark 1 (Convergence Analysis of FL): According to existing work [6], we provide the convergence analysis on the FL of the proposed algorithm by some adjustments. More details are shown as follows.

Proof: For theoretical analysis, the assumptions of the local loss function are listed as below:

Assumption 1 We assume the following for the n-th MD:

1) The local loss function $F_n(w)$ of the n-th MD is convex.
2) $F_n(w)$ is ρ-Lipschitz, i.e., $||F_n(w) - F_n(w')|| \leq \rho||w - w'||$ for any w, w'.

TABLE I
PARAMETERS IN SIMULATIONS

Nations	Values
The number of MDs N	10
The number of channels L	5
Channel bandwidth W	0.1 MHz
The number of CPU cycles C per unit data sampling	$[10^{10}, 1.9 \times 10^{10}]$ cycles/unit
Each iteration duration τ	10 s
Computation capacity f_n of the MD	$[2 \times 10^9, 4 \times 10^9]$ cycles/s
The size of local parameter for each MD	10^6 bit
The effective capacitance parameter α	10^{-28}
The coefficient determined by machine learning model ζ	1
The upper limit value of the average energy outage $\Pr_n^{\text{th}}$	4%

3) $F_n(w)$ is β-smooth, i.e., $||\nabla F_n(w) - \nabla F_n(w')|| \leq \beta||w - w'||$ for any w, w'.

We also define the following metric to capture the divergence between the gradient of a local loss function of the n-th MD and the gradient of the global loss function, which is a gradient-weighted aggregation of the local loss functions of all MDs.

For any n and w, we define δ_n as an upper bound of $||\nabla F_n(w) - \nabla F_n(w')|| \leq \delta_n$. And we also define $\delta(t) \triangleq \frac{\sum_{n=1}^{N} b_n(t)\delta_n}{\sum_{n=1}^{N} b_n(t)}$.

Following [6], after T global aggregations, when the learning rate $\eta < \frac{1}{\beta}$, we have the bound between the global loss function $F(w^T)$ after T global aggregations and the optimal global loss function $F(w^*)$,

$$F(w^T) - F(w^*) \leq \frac{1}{2\eta\varphi T} + \sqrt{\frac{1}{4\eta^2\varphi^2 T^2} + \frac{\rho H^*(\varrho)}{\eta\varphi\varrho}} + \rho H^*(\varrho), \quad (56)$$

where $\varphi = \omega(1 - \frac{\beta\eta}{2})$, $\omega = \min_t \frac{1}{||w^{t-1} - w^*||}$, $H^*(\varrho) = \max_t \frac{\delta(t)}{\beta}((\eta\beta + 1)^\varrho - 1) - \eta\delta(t)\varrho$ and ϱ is the same number of local gradient descent for all MDs. ■

V. SIMULATION AND DISCUSSION

In this section, we evaluate the performance of the proposed algorithm with extensive simulations.[5] Considering all MDs are randomly distributed in a fixed region. We set the bandwidth of channel between each MD and the edge server as 0.1 MHz. The number of CPU cycles C for each MD to perform local model training of unit data sampling takes range from 10^{10} cycle/unit to $1.9{*}10^{10}$ cycle/unit. The simulation parameters are detailed in Table I.

[5]The source code used in the numerical evaluations detailed in this manuscript is available online at https://github.com/chushunfeng/code_TCCN.

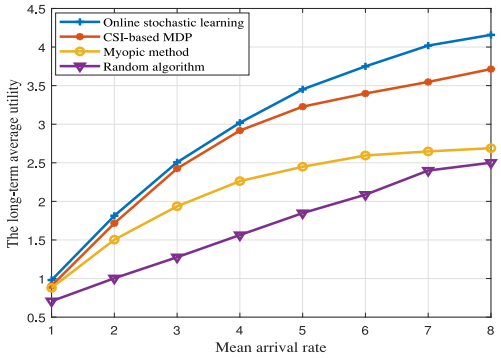


Fig. 2. The long-term average utility $\mathcal{U}$ (MB) v.s. the mean arrive rate λ (J) of the random new arrived energy with $E^{\max} = 6$ J.

We compare our proposed online stochastic learning algorithm with three other benchmark algorithms. One is the CSI-based MDP algorithm, where the edge server takes corresponding decisions based on the channel state only at the current iteration so as to optimize the average utility of all MDs. The second benchmark algorithm is a myopic method, which only considers the current utility. For myopic method, the edge server never considers long-term utilities. The last benchmark algorithm is a random resource scheduling method, where the edge server takes random actions in the feasible regions. The performance of the proposed algorithm is evaluated by averaging over 5000 experiments.

Fig. 2 illustrates the long-term average utility $\mathcal{U}$ v.s. the mean arrival rate λ of the random new arrived energy with $E^{\max} = 6$ J. It can be observed that the performance of the proposed online stochastic learning algorithm outperforms other benchmark algorithms for all the investigated average arrival rate λ. When the value of the mean arrival rate λ is relatively small, the performance of online learning is close to that of benchmark algorithms, especially the CSI-based MDP algorithm. The reason is twofold. First, small λ will result in a limited energy level in the battery of the MD. Due to insufficient battery energy, the MDs are constrained within a small action space compared with those with sufficient battery power. The second reason is that a large amount of battery energy is used for uploading local parameters. When the battery energy of MD is insufficient, the energy for local training of FL is small, which leads to small long-term average utility. In contrast, when the battery level is high, MD's actions will become diverse, and more energy will be used for local training in FL process.

Fig. 3 illustrates the long-term average utility $\mathcal{U}$ v.s. the maximum battery capacity $E^{\max}$ of MD. We observe that the long-term average utility $\mathcal{U}$ increases approximately linearly as the maximum battery capacity $E^{\max}$ of MD increases in all algorithms, Fig. 4 shows the impact of the number of CPU cycles for MD to perform local training of unit data sampling C on the long-term average utility $\mathcal{U}$. It is obvious that the long-term average utility decreases with the number of CPU cycles for unit training data sampling C. In addition, as the number of CPU cycles for unit training data sampling C continues to increase, the performance gaps among different

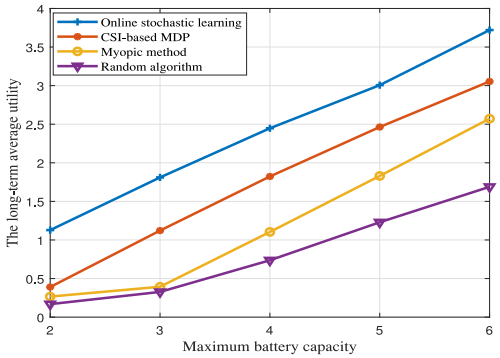


Fig. 3. The long-term average utility $\mathcal{U}$ (MB) v.s. the maximum battery capacity $E^{\max}$ (J) of the MD.

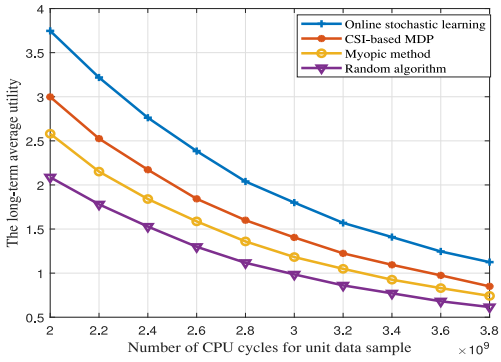


Fig. 4. The long-term average utility $\mathcal{U}$ (MB) v.s. the number of CPU cycles for MD to perform local model training of unit data sampling C (cyc/MB).

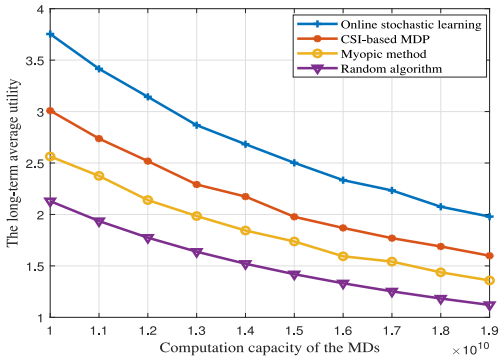


Fig. 5. The long-term average utility (MB) $\mathcal{U}$ v.s. the computation capacity f (Hz) of MD.

algorithms also decrease. Fig. 5 depicts the long-term average utility $\mathcal{U}$ versus the computation capacity f of MD. Intuitively, the MDs with more computing capacity will lead to a higher long-term average utility $\mathcal{U}$. However, the opposite is true, it is caused by (3) and the limited energy of MD in each iteration. In other words, a more powerful computing capacity requires more computing energy. Due to the limited amounts of energy available to MDs in each iteration, MDs can only reduce the size of sampled data used for local training.

Fig. 6 describes the relationship among the wireless channel state, the energy state of MD and the optimal transmit

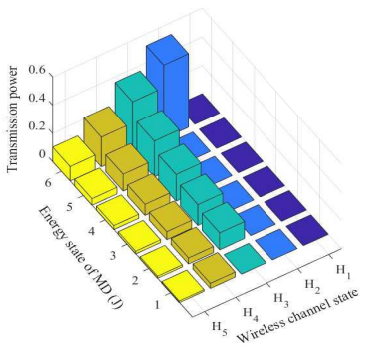


Fig. 6. The transmission power (W) v.s. the wireless channel state and the energy state of MD.

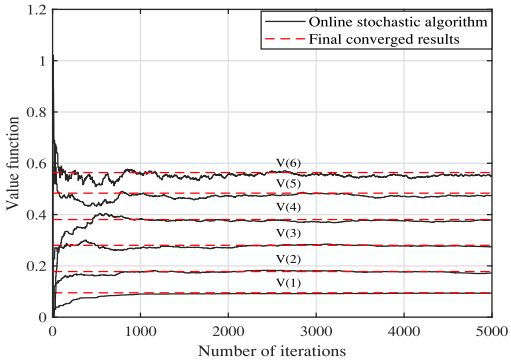


Fig. 7. Convergence property of the proposed online stochastic learning algorithm.

power policy. In our simulation settings, H_1 represents the worst channel state, while H_5 represents the best channel state in our system. From the figure, we can see that the MD avoids data transmission to save battery energy when the MD is in a very poor channel state (H_1). In addition, we find that for a given channel state, the optimal transmit power monotonously increases with the energy state of the MD.

Fig. 7 illustrates the convergence property of the proposed distributed online learning algorithm. It can be seen that the online stochastic algorithm converges quite fast and after 1500 iterations, the values are close to the final converged results. Moreover, it is clear that the value functions quickly approach the final converged results when the number of iterations grows.

In Figs. 8 and 9, we show how the FL accuracy and the loss value changes as the number of iterations varies on FashionMNIST data. From Figs. 8 and 9, we can see that our proposed algorithm is significantly better than the baseline algorithms, and its convergence speed is also significantly faster than the baseline algorithms. Furthermore, our proposed approach has less volatility than the benchmark algorithms due to the larger amount of training data of MDs. Figs. 10 and 11 show how the FL accuracy and the loss value changes with the number of iterations on Cifar-10 data. We can also see that, the proposed FL algorithm can achieve up to 5% gains in terms of the accuracy compared with the baseline algorithms. Furthermore, the loss value of our proposed scheme is

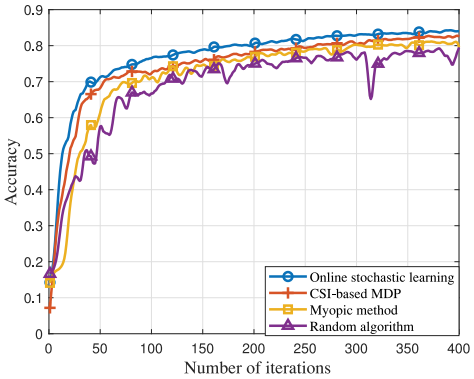


Fig. 8. The FL accuracy v.s. number of iterations on FashionMNIST data.

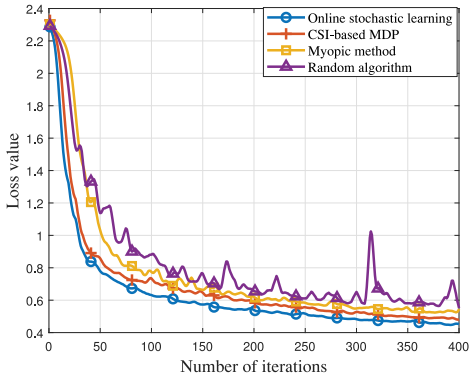


Fig. 9. The FL loss value v.s. number of iterations on FashionMNIST data.

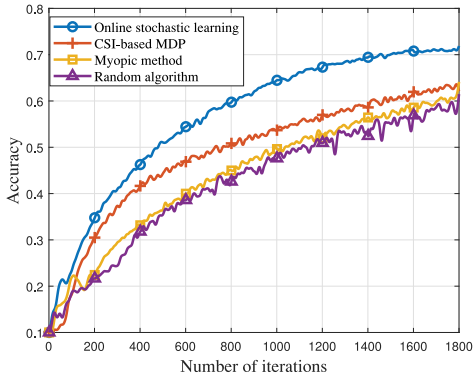


Fig. 10. The FL accuracy v.s. number of iterations on Cifar-10 data.

also reduced by more than 10% compared with the benchmark algorithms. As seen from Fig. 8 to 10, the random algorithm that randomly selects local data for training has lowest learning performance and speed, as well as poor stability owing to huge fluctuations compared with other algorithms.

VI. Conclusion

In this paper, we study a CMDP problem of FL with a MEC sever, where the MDs send local model updates trained on their local sensitive data iteratively to the edge server, and then the edge server aggregates the parameters from MDs

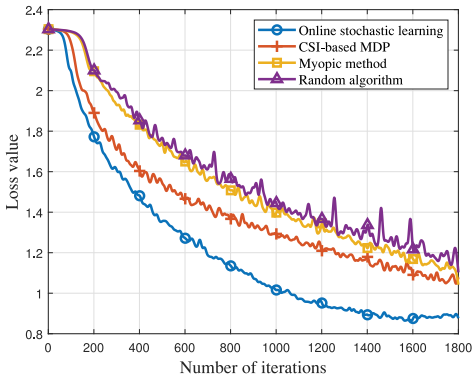


Fig. 11. The FL loss value v.s. number of iterations on Cifar-10 data.

and broadcasts the aggregated parameters to MDs. We first model the resource scheduling problem in the synchronous FL process as a CMDP problem, and we use the size of the training samples as the performance of FL for analysis. Due to the coupling between iterations and the complexity of the state-action space, we cannot directly solve the CMDP problem. Thus, we analyze the problem by equivalent Bellman equations and use approximate MDP and stochastic learning methods to simplify the CMDP problem so as to approximate the state value function. Then, we design static algorithm to obtain the static policy for each MD based the approximate state value function. Finally, we provide theoretical analysis for the convergence of the online stochastic learning algorithm. The simulation results show that the performance of the stochastic leaning outperforms other benchmark schemes.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018.

[3] K. Wei *et al.*, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[4] K. Wei *et al.*, "User-level privacy-preserving federated learning: Analysis and performance optimization," *IEEE Trans. Mobile Comput.*, vol. 21, no. 9, pp. 3388–3401, Sep. 2022.

[5] C. Ma *et al.*, "On safeguarding privacy and security in the framework of federated learning," *IEEE Netw.*, vol. 34, no. 4, pp. 242–248, Jul./Aug. 2020.

[6] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[7] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2020.

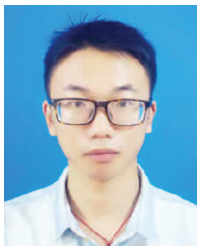
[8] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," 2016, *arXiv:1602.05629*.

[9] M. Li, D. G. Andersen, A. J. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2014, pp. 19–27.

[10] D. Alistarh, D. Grubic, J. Z. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1709–1720.

[11] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," 2017, *arXiv:1712.01887*.

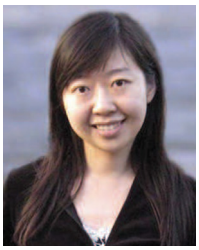
[12] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Sparse binary compression: Towards distributed deep learning with minimal communication," in *Proc. Int. Joint Conf. Neural Netw.*, 2019, pp. 1–8.
[13] T. T. Anh, N. C. Luong, D. Niyato, D. I. Kim, and L.-C. Wang, "Efficient training management for mobile crowd-machine learning: A deep reinforcement learning approach," *IEEE Wireless Commun. Lett.*, vol. 8, no. 5, pp. 1345–1348, Oct. 2019.
[14] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.
[15] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1146–1159, Feb. 2020.
[16] S. Ha, J. Zhang, O. Simeone, and J. Kang, "Coded federated computing in wireless networks with straggling devices and imperfect CSI," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2019, pp. 2649–2653.
[17] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.
[18] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1387–1395.
[19] U. Mohammad and S. Sorour, "Adaptive task allocation for asynchronous federated and parallelized mobile edge learning," 2019, *arXiv:1905.01656*.
[20] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.
[21] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient radio resource allocation for federated edge learning," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2020, pp. 1–6.
[22] R. Jin, X. He, and H. Dai, "Communication efficient federated learning with energy awareness over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5204–5219, Jul. 2022.
[23] J. Kim, D. Kim, J. Lee, and J. Hwang, "A novel joint dataset and computation management scheme for energy-efficient federated learning in mobile edge computing," *IEEE Wireless Commun. Lett.*, vol. 11, no. 5, pp. 898–902, May 2022.
[24] Y. Mao, J. Zhang, and K. B. Letaief, "Grid energy consumption and QoS tradeoff in hybrid energy supply wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3573–3586, May 2016.
[25] Y. Mao, J. Zhang, and K. B. Letaief, "A Lyapunov optimization approach for green cellular networks with hybrid energy supplies," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2463–2477, Sep. 2015.
[26] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.
[27] X. Wu, J. Li, M. Xiao, P. C. Ching, and H. V. Poor, "Multi-agent reinforcement learning for cooperative coded caching via homotopy optimization," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5258–5272, Aug. 2021.
[28] P. Wu, J. Li, L. Shi, M. Ding, K. Cai, and F. Yang, "Dynamic content update for wireless edge caching via deep reinforcement learning," *IEEE Commun. Lett.*, vol. 23, no. 10, pp. 1773–1777, Oct. 2019.
[29] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-IID data with reinforcement learning," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 1698–1707.
[30] D. Van Le and C. K. Tham, "Quality of service aware computation offloading in an ad-hoc mobile cloud," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8890–8904, Sep. 2018.
[31] X. Zhao, W. Chen, J. Lee, and N. B. Shroff, "Delay-optimal and energy-efficient communications with Markovian arrivals," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1508–1523, Mar. 2020.
[32] A. Biason, S. Dey, and M. Zorzi, "A decentralized optimization framework for energy harvesting devices," *IEEE Trans. Mobile Comput.*, vol. 17, no. 11, pp. 2483–2496, Nov. 2018.
[33] S. Knorn, S. Dey, A. Ahlén, and D. E. Quevedo, "Distortion minimization in multi-sensor estimation using energy harvesting and energy sharing," *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2848–2863, Jun. 2015.
[34] N. Sharma, N. Mastronarde, and J. Chakareski, "Accelerated structure-aware reinforcement learning for delay-sensitive energy harvesting wireless sensors," *IEEE Trans. Signal Process.*, vol. 68, no. 1, pp. 1409–1424, Feb. 2020.
[35] Y. Cui and V. K. N. Lau, "Distributive stochastic learning for delay optimal OFDMA power and subband allocation," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4848–4858, Sep. 2010.
[36] Y. Cui, V. K. N. Lau, R. Wang, H. Huang, and S. Zhang, "A survey on delay-aware resource control for wireless systems—Large deviation theory, stochastic Lyapunov drift, and distributed stochastic learning," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1677–1701, Mar. 2012.
[37] X. Zhao and W. Chen, "Non-orthogonal multiple access for delay-sensitive communications: A cross-layer approach," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 5053–5068, Jul. 2019.
[38] M.-L. Ku, W. Li, Y. Chen, and K. J. R. Liu, "On energy harvesting gain and diversity analysis in cooperative communications," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2641–2657, Dec. 2015.
[39] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2014, pp. 661–670.
[40] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 443–461, 3rd Quart., 2011.
[41] L. Lei, Y. Kuang, N. Cheng, X. S. Shen, Z. Zhong, and C. Lin, "Delay-optimal dynamic mode selection and resource allocation in device-to-device communications—Part I: Optimal policy," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3474–3490, May 2016.
[42] I. Bettesh and S. S. Shamai, "Optimal power and rate control for minimal average delay: The single-user case," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 4115–4141, Sep. 2006.
[43] R. Wang and V. K. N. Lau, "Delay-aware two-hop cooperative relay communications via approximate MDP and stochastic learning," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7645–7670, Nov. 2013.
[44] V. S. Borkar, "Stochastic approximation with two time scales," *Syst. Control Lett.*, vol. 29, pp. 291–294, Feb. 1997.
[45] V. S. Borkar, "An actor-critic algorithm for constrained Markov decision processes," *Syst. Control Lett.*, vol. 54, no. 3, pp. 207–213, Mar. 2005.
[46] L. Ljung, G. Pflug, and H. Walk, *Stochastic Approximation and Optimization of Random Systems*. Cambridge, MA, USA: Birkhäuser, 2012.
[47] L. Lei, Y. Kuang, N. Cheng, X. S. Shen, Z. Zhong, and C. Lin, "Delay-optimal dynamic mode selection and resource allocation in device-to-device communications—Part II: Practical algorithm," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3491–3505, May 2016.



Shunfeng Chu received the M.S. degree from the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2018, where he is currently pursuing the Ph.D. degree. His research interests include federated learning, dynamic programming, and game theory.

Jun Li (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009. From January 2009 to June 2009, he worked with the Department of Research and Innovation, Alcatel Lucent Shanghai Bell as a Research Scientist. From June 2009 to April 2012, he was a Postdoctoral Fellow with the School of Electrical Engineering and Telecommunications, the University of New South Wales, Australia. From April 2012 to June 2015, he was a Research Fellow with the School of Electrical Engineering, the University of Sydney, Australia. Since 2015, He has been a Professor with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. He was a visiting professor with Princeton University from 2018 to 2019. He has coauthored more than 200 papers in IEEE journals and conferences and holds one U.S. patents and more than ten Chinese patents in these areas. His research interests include network information theory, game theory, distributed intelligence, multiple agent reinforcement learning, and their applications in ultra-dense wireless networks, mobile edge computing, network privacy and security, and industrial Internet of Things. He is serving as an Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATION and the TPC member for several flagship IEEE conferences.

Jianxin Wang received the M.Sc. and Ph.D. degrees in electronic and information engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1987 and 1999 respectively. Since 1987, he has been with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology as a Teaching Assistant, a Lecturer, and an Associate Professor and since 2001 as a Professor. He was a Visiting Researcher with the Institute of Telecommunications, University of Stuttgart, Germany, in 2000, December 2012, and January 2016, respectively. His main research areas are communications signal processing and software defined radio.



Zhe Wang (Member, IEEE) received the Ph.D. degree in electrical engineering from The University of New South Wales, Sydney, Australia, in 2014. From 2014 to 2020, she was a Research Fellow with The University of Melbourne, Australia, and the Singapore University of Technology and Design, Singapore. She is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. Her research interests include applications of optimization, game theory, and machine learning to resource allocation in communications and networking.

Ming Ding (Senior Member, IEEE) received the B.S. and M.S. degrees (with first-class Hons.) in electronics engineering and the Ph.D. degree in signal and information processing from Shanghai Jiao Tong University, Shanghai, China, in 2004, 2007, and 2011, respectively. From April 2007 to September 2014, he worked with the Sharp Laboratories of China, Shanghai, China as a Researcher/Senior Researcher/Principal Researcher. He is Currently a Senior Research Scientist with Data61, CSIRO, Sydney, NSW, Australia. He has authored more than 150 papers in IEEE journals and conferences, all in recognized venues, and around 20 3GPP standardization contributions, as well as two books, i.e., *Multi-Point Cooperative Communication Systems: Theory and Applications* (Springer, 2013) and *Fundamentals of Ultra-Dense Wireless Networks* (Cambridge University Press, 2022). Also, he holds 21 U.S. patents and has co-invented another 100+ patents on 4G/5G technologies. His research interests include information technology, data privacy and security, and machine learning and AI. He is Currently an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. Besides, he has served as a Guest Editor/Co-Chair/Co-Tutor/TPC Member for multiple IEEE top-tier journals/conferences and received several awards for his research work and professional services.

Yijin Zhang (Senior Member, IEEE) received the Ph.D. degree in information engineering from the Chinese University of Hong Kong in 2010. He joined the Nanjing University of Science and Technology, China, in 2011, where he is currently a Professor with the School of Electronic and Optical Engineering. His research interests include sequence design and resource allocation for communication networks.

Yuwen Qian received the Ph.D. degree in automatic engineering from the Nanjing University of Science and Technology, Nanjing, China, in 2011. From July 2002 to June 2011, he was a Lecturer of Automation School with the Nanjing University of Science and Technology, where he has been an Associate Professor with the School of Electronic and Optical Engineering since May 2019.

Wen Chen (Senior Member, IEEE) is a Tenured Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University, China, where he is the Director of Broadband Access Network Laboratory. He has published more than 110 papers in IEEE journals and more than 120 papers in IEEE Conferences, with citations more than 8000 in google scholar. His research interests include multiple access, wireless AI, and metasurface communications. He is the Shanghai Chapter Chair of IEEE Vehicular Technology Society, the Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE ACCESS, and IEEE OPEN JOURNAL OF VEHICULAR TECHNOLOGY. He is a Fellow of the Chinese Institute of Electronics and the distinguished lecturers of IEEE Communications Society and IEEE Vehicular Technology Society.